Cross-Domain Authorship Verification Based on Topic Agnostic Features

Oren Halvani *, Lukas Graner, and Roey Regev

Fraunhofer Institute for Secure Information Technology SIT
Rheinstrasse 75, 64295 Darmstadt, Germany
{FirstName.LastName}@SIT.Fraunhofer.de

Abstract Authorship verification (AV) is a research branch in digital text forensics that deals with the problem to determine whether two documents were written by the same author. Research activities in the context of AV have steadily increased in recent years, which have led to a variety of approaches trying to solve this problem. Many of these approaches, however, make use of features that are related to or influenced by the topic of the documents. Therefore, it may accidentally happen that their verification results are based not on the writing style alone (the actual focus of AV), but on the topic of the documents. To address this problem, we propose in the context of the AV shared task at the PAN 2020 workshop an alternative approach, which considers only topic-agnostic features in its classification decision. On the official test set, our approach was ranked third out of all submitted approaches.

1 Introduction

With the constant increase of documents worldwide, more and more possibilities of identity misuse are becoming established. One example of such identity abuse is "CEO Fraud" – a sophisticated email scam – in which an attacker sends an email to an employee on behalf of a CEO to perform a specific action (e. g., transferring money or sending confidential company information). Another form of identity abuse occurs in the context of compromised accounts, where the attacker distributes messages in the name of the victim. In addition, identity abuse can occur in fake reviews in which, for example, an attempt is made on behalf of an alleged person to positively advertise a product or service provider. A countermeasure regarding these scenarios is to compare the writing style of the questioned documents with the writing style of those documents for which the true author $\mathcal{A}$ is known. By this, the question can be answered (with a certain degree of probability) whether the unknown document was also written by $\mathcal{A}$. The comparison of documents based on their writing style is particularly relevant if no other metadata are available to clarify the identity of the unknown author. Authorship verification (AV), which is a branch of digital text forensics, has been dealing with this question for over two decades. Technically, AV represents a similarity detection

* Corresponding author.

problem, where for an unknown document $\mathcal{D}_{\mathcal{U}}$ and a known document $\mathcal{D}_{\mathcal{A}}$ it has to be determined whether both were written by the same author $\mathcal{A}$. The focus of the similarity determination in the context of AV is on the writing style and not on other factors such as the topic or genre. Therefore, if $\mathcal{D}_{\mathcal{U}}$ and $\mathcal{D}_{\mathcal{A}}$ share the same topic but were written by different authors, a naive AV method might erroneously assume a high degree of similarity, resulting in a clear failure to achieve its intended goal.

A large number of existing AV methods including [4,6,21,22,25,26] make use of character n-grams (overlapping character sequences), which are known to be closely associated to particular content words and, therefore, can be problematic when dealing with authorship [20]. Style analysis, however, must abstract from content and focus on content-independent formal properties of linguistic expressions in a text [9]. In the light of this conclusion, we propose an alternative approach which, by design, considers only such text units that reflect valid stylistic markers. Our contribution in this paper is twofold: First, we propose a number of topic-agnostic feature categories that effectively quantify the writing style of documents. Second, we propose a transparent AV method that can be applied to challenging AV tasks. These include cases, where $\mathcal{D}_{\mathcal{U}}$ and $\mathcal{D}_{\mathcal{A}}$ consist of only a few sentences or cases where both differ thematically.

The rest of the paper[1] is organized as follows. Section 2 discusses previous work in the context of AV. In Section 3, we propose a number of feature categories, which will be used by our AV method introduced in Section 4. Afterwards, we present our experimental evaluation in Section 5 and, finally, in Section 6 we conclude the work and provide ideas for future work.

2 Previous Work

The core of every AV method is a classification model that aims to decide whether a questioned document $\mathcal{D}_{\mathcal{U}}$ was written by a certain author $\mathcal{A}$, for which a set $\mathbb{D}_{\mathcal{A}} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots\}$ of reference documents is given. With regard to their classification models, we have identified three categories of AV methods in our previous research work [13], which are summarized below.

The first category are **unary** AV methods that determine their classification model solely on the basis of $\mathbb{D}_{\mathcal{A}}$. A unary AV method assumes $\mathcal{D}_{\mathcal{U}}$ to be written by $\mathcal{A}$, if it is stylistically similar to the documents in $\mathbb{D}_{\mathcal{A}}$. The second category are **binary-intrinsic** AV methods that determine their classification model on the basis of a given training corpus. This corpus consists of a number of verification cases with a ground truth regarding the classes `Y` (same-author) and `N` (different-author). A binary-intrinsic AV method treats the unknown and known documents as a single unit X (for example, a feature vector). If X is more similar to the `Y`-cases, the method accepts $\mathcal{A}$ as the author of $\mathcal{D}_{\mathcal{U}}$. If, on the other hand, X is more similar to the `N`-cases, $\mathcal{D}_{\mathcal{U}}$ is assumed to be written by another author. In any case, the decision is made solely on the basis

[1] Portions of this paper are based on our published work [12]. We therefore kindly ask the interested reader who would like to cite this article to use this reference. Note that a video presentation of our approach is available at `http://bit.ly/TAVeer`

of X and the learned model (hence, intrinsic). The third category are **binary-extrinsic** AV methods that determine their classification model on the basis of external (so-called *impostor* [21]) documents which, for example, are gathered by using a search engine. In this context, the documents in $\mathbb{D}_{\mathcal{A}}$ represent samples of the true author $\mathcal{A}$, while the impostor documents act as samples of an author different than $\mathcal{A}$. Binary-extrinsic AV methods assume $\mathcal{D}_{\mathcal{U}}$ to be written by $\mathcal{A}$, if it is stylistically similar to the documents in $\mathbb{D}_{\mathcal{A}}$. However, if $\mathcal{D}_{\mathcal{U}}$ is more similar to the impostor documents, it is assumed to have been written by an author other than $\mathcal{A}$.

Over the last two decades, numerous AV approaches have been proposed that can be assigned to one of the above categories. An approach that we refer to as AVIF and that belongs to the category of unary AV methods was developed by Neal et al. [22] for the purpose of continuous verification. Their method is based on an isolation forest classifier, which, like many other AV methods, considers character n-grams as underlying features. AVIF achieved a high recognition accuracy using very small training samples of 50 and 100-character blocks. However, in their study the authors explain that the method was only evaluated on positive samples (in other words, instances of the `Y`-class). Therefore, it is not clear how well AVIF performs under realistic conditions where both cases (`Y` **and** `N`) are present.

A well-known binary-intrinsic AV approach, which we denote by the name ProfAV, was proposed by Potha and Stamatatos [24]. Their method considers two documents $\mathcal{D}_{\mathcal{U}}$ and $\mathcal{D}_{\mathcal{A}}$ as character n-gram profiles and measures their relative differences using a predefined dissimilarity function. If the resulting dissimilarity score exceeds a certain threshold (derived from the distribution of `Y`/`N`-samples in a given training corpus), $\mathcal{D}_{\mathcal{U}}$ is assumed to be written by $\mathcal{A}$. Potha and Stamatatos [24] demonstrated that ProfAV was able to outperform every single AV method submitted to the first AV-competition as a part of the PAN shared tasks [16].

One of the most influential and successful binary-extrinsic AV approach is the *Impostors Method* (IM) proposed by Koppel and Winter [21], which laid the foundations for many subsequent AV approaches including [28,17,18,19,25]. IM can be broken down into two steps. First, appropriate impostor documents have to be collected according to a predefined strategy (for example, using a search engine). In the second step, a feature selection technique based on character n-grams is applied iteratively to measure the similarity between pairs of documents. If, given this measure, a suspect is picked out from among the impostor set with sufficient salience, then the suspect is assumed to be the author of $\mathcal{D}_{\mathcal{U}}$ [21]. The IM variants of Khonji and Iraqi [17] and Seidman [28] were the best-performing approaches in the first and second PAN-AV competitions [16,29]. Another strong approach that belongs to the category of binary-extrinsic AV methods is the so-called NNCD method proposed by Veenman and Li [32]. In contrast to IM, their method delegates the entire feature engineering procedure to a state of the art compression-algorithm. Here, $\mathcal{D}_{\mathcal{U}}$ is assumed to be written by $\mathcal{A}$ if the compressed representation of $\mathcal{D}_{\mathcal{U}}$ is dissimilar to those of the impostor documents. Both NNCD [32] and GenIM [28] were the best-performing approaches in the first PAN AV competition [16].

3 Feature Categories

In this section, we propose a number of feature categories that are used by our AV approach to capture the writing style of documents. A part of these are derived from certain feature categories used in previous studies. The remaining feature categories, however, have been not considered so far in the context of AV, at least to our best knowledge. All feature categories are summarized in Table 1 along with a number of examples. In the following subsections, we first introduce all feature categories in detail. Afterwards, we explain which design decisions we made in regard to their hyperparameters. Finally, we describe the scope from where all proposed features are extracted and how we normalized them.

ID	Feature category	Sample output	Range
F_{1-3}	Punctuation n-grams	{(`,`, `.`)}	$n \in \{1,2,3\}$
F_4	TA sentence and clause starters	{(`however`), (`, there`)}	—
F_5	TA sentence endings	{(`this`)}	—
F_{6-9}	TA token n-grams	{(`however , there`), (`, there is`), (`there is an`), (`to this .`)}	$n \in \{1,2,3,4\}$
F_{10-11}	TA masked token n-grams	{(`is an #`), (`# to this`)}	$n \in \{3,4\}$

Table 1. All 11 feature categories considered by TAVeer (feature categories with the TA-prefix are proposed by us). The third column shows the output for the sample sentence: `"However, there is an opposing view to this."` Note that for the n-gram-based feature categories, each setting of n results in an individual feature category.

3.1 Topic-Agnostic Words and Phrases

Function words can be seen as the most common choice in the field of authorship analysis, when it comes to select topic-agnostic features. However, in the literature it often remains unclear what is exactly understood and represented under the term "function words". In many existing studies (for example, [7,15,34]) no detailed explanation is provided regarding the question, which specific function word categories (or at least which specific words) were taken into account. Another peculiarity that can be seen in the literature, is the varying number of considered function words. For example, Chandrasekaran [7], Binongo [3], Srinivasa [27] and Zhao and Zobel [33] make use of 24, 50, 150 and 365 function words, respectively. In view of these different numbers, the question arises why only individual subsets are considered rather than using the entire spectrum of function words. Instead of making use of non-structured and incomplete lists, Varela et al. [31] and Pavelec et al. [23] follow a different approach, in which they consider subcategories of function words such as pronouns, conjunctions, subclasses of adverbs and other word forms. By this, a better insight can be gained regarding the question which specific types of function words were actually taken into account.

Motivated by this idea, we opted for a similar but more systematic approach, in which we consider all existing categories of function words along with other carefully selected topic-agnostic (hereafter, abbreviated as **TA**) categories. First, we assemble a

comprehensive list $\mathcal{L}_{\text{TA}}$ consisting of words and phrases that belong to these categories (cf. Table 2). Based on $\mathcal{L}_{\text{TA}}$, we then derive different TA feature categories (described below) that can be used to model the writing style of documents across different linguistic layers. For the construction of $\mathcal{L}_{\text{TA}}$, we use a variety of words and phrases classified into 20 categories including function words, empty verbs, contractions, generic adverbs as well as transitional words and phrases. All considered words and phrases, which are known in the literature [23,3,30] to be content and topic independent, have been collected from different sources, in particular, linguistic books and stylometry papers. The transitional phrases cover a number of categories including *causation*, *contrast*, *similarity*, *clarification*, *conclusion*, *purpose* and *summary*. With regard to the verbs, we also take the respective tenses[2] into account (for example, `give` $\rightarrow$ {`gives, giving, gave, given`}) in order to enrich $\mathcal{L}_{\text{TA}}$. All categories of words and phrases contained in $\mathcal{L}_{\text{TA}}$ are summarized in Table 2 along with a number of examples. Note that due to the ambiguities occurring in the English language, a number of function words appear in multiple categories. For example, `"but"` and `"for"` are both prepositions and conjunctions, whereas `"few"` represents a pronoun and a quantifier. However, regarding the features in $\mathcal{L}_{\text{TA}}$, we do not differentiate between the different meanings of these homographs[3]. Based on $\mathcal{L}_{\text{TA}}$, we derive additional feature categories which are described in the following.

Category	Examples
Conjunctions	{and, as, because, but, either, for, hence, however, if, neither, ...}
Determiners	{a, an, both, each, either, every, no, other, our, some, ...}
Prepositions	{above, after, among, below, beside, between, beyond, inside, ...}
Pronouns	{all, another, any, anyone, anything, everything, few, he, her, ...}
Quantifiers	{any, certain, each, either, few, less, lots, many, more, most, ...}
Auxiliary verbs	{can, could, might, must, ought, shall, will, ...}
Delexicalised verbs	{get, go, take, make, do, have, give, set, ...}
Empty verbs	{do, did, does, got, have, had, had, gives, giving, gave, ...}
Helping verbs	{am, is, are, was, were, be, been, will, should, would, could, ...}
Contractions	{i'm, i'd, i'll, i've, he's, it's, we'd, she's, it'll, we're, ...}
Adverbs of degree	{almost, enough, hardly, just, nearly, quite, simply, so, too, ...}
Adverbs of frequency	{again, always, never, normally, rarely, seldom, sometimes, ...}
Adverbs of place	{below, everywhere, here, in, inside, into, nowhere, out, ...}
Adverbs of time	{already, during, immediately, just, recently, still, then, yet, ...}
Pronominal adverbs	{hereafter, hereby, thereafter, thereby, therefore, therein, ...}
Focusing adverbs	{especially, mainly, particularly, generally, only, simply, ...}
Conjunctive adverbs	{likewise, meanwhile, moreover, namely, nonetheless, otherwise, ...}
Transition words	{besides, furthermore, generally, hence, thus, however, ...}
Transitional phrases	{of course, as a result, in addition, because of, in contrast, ...}
Phrasal prepositions	{as opposed to, in regard to, in relation to, inspite of, out of, ...}

Table 2. All categories of TA-based words and phrases. The list $\mathcal{L}_{\text{TA}}$ is created by taking the union of all categories.

[2] For this we used *pattern* [8] available at https://github.com/clips/pattern.

[3] Homographs are words with the same spelling but different meaning.

Punctuation n-Grams (F_{1-3}) Punctuation marks represent syntactic features that quantify the grammatical structures an author uses and, thus, are content and topic independent [30]. As punctuation n-grams we define a sequence of consecutive punctuation marks where letters, digits and other non-punctuation characters are skipped (cf. Table 1). Among others, punctuation n-grams capture specific symbols that occur at word-internal level such as hyphens or apostrophes used in contractions (e. g., `we've` or `they're`). Furthermore, they allow to recognize unusual punctuation habits reflecting the individual writing style of an author such as combinations of question and exclamation marks (e. g., **`?!?`** or **`!?!`**), which occur in informal documents. In total, we consider three punctuation n-gram feature categories (F_{1-3}) that are not dependent on the list $\mathcal{L}_{\text{TA}}$. However, the feature categories F_{6-11} make use of F_1 (punctuation unigram).

TA Sentence and Clause Starters (F_4) Words or phrases that appear at the beginning of sentences or clauses can reflect one aspect of an author's writing style. We therefore consider such sentences and clause starters as a distinct feature category. However, since our focus lies on TA-based features, we make sure that a word or phrase appearing at the beginning of a sentence or a clause is included in $\mathcal{L}_{\text{TA}}$. Note that in case of clauses, we consider the preceding punctuation mark (comma or semicolon) together with the subsequent word or phrase as a whole feature (cf. Table 1).

TA Sentence Endings (F_5) Words or phrases that appear at the end of sentences might also reflect a stylistic habit of authors. We therefore consider such features as a distinct feature category and make sure (analogous to F_4) that they are included in $\mathcal{L}_{\text{TA}}$.

TA Token n-Grams (F_{6-9}) These feature categories are a form of standard token n-grams with the restriction that each token t_i in a token n-gram $(t_1, t_2, \ldots, t_n)$ represents either a punctuation or a word appearing in $\mathcal{L}_{\text{TA}}$ (cf. Table 1). Note that for $n = 1$, the respective feature category F_6 is essentially the list $\mathcal{L}_{\text{TA}}$, which is obtained by merging all categories listed in Table 2.

TA Masked Token n-Grams (F_{10-11}) These feature categories also represent a form of token n-grams with the restriction that $n-1$ tokens in a token n-gram $(t_1, t_2, \ldots, t_n)$ are either punctuation marks or words appearing in $\mathcal{L}_{\text{TA}}$. The remaining $n - 2$ tokens, on the other hand, represent topic-related words, which are then **masked** by the non-punctuation character `#`. The intention here is to enable the detection of contexts surrounding or adjacent to topic-agnostic words (cf. Table 1).

3.2 Feature Category Ranges

In previous AV works (e. g., [24,14,4]) n-gram-based feature categories have been treated as a single concept, where the most suitable n was chosen on the basis of a hyperparameter optimization procedure. In contrast to this, we treat n-gram-based feature categories **independently** so that, for example, punctuation 2- and 3-grams represent

two individual feature categories. There is a simple justification for this decision: If we would restrict ourselves to a specific n optimized on a training corpus, we might miss important features occurring in the unseen data (test corpus) that can only be captured with an alternative setting of n. Allowing multiple settings of n for the same feature category can therefore help counteract a mismatch of existing features between training and test data.

In the following, we explain the considerations behind the ranges of the n-gram-based feature categories listed in Table 1. For the punctuation n-grams, we set $n = 1$ as a lower limit which is useful in cases where sentences comprise only a single punctuation (e. g., full-stop, question or exclamation mark). As an upper limit, we set $n = 3$, as it can be expected that longer punctuation sequences between the unknown and known documents will be scarce (more on this in the next subsection). Regarding TA token n-grams, we set $n = 1$ and $n = 4$ as a lower and upper limit, respectively. For the former, we aim to capture at least single words in the documents. Here, we expect that a part of these features will be present in both documents, in most of the cases. With regard to longer sequences, we aim to capture specific phrases that can be relevant for individual authors. However, sequences with more than four tokens are less likely to appear, especially between short documents so that $n = 4$ can be seen as a good compromise. For the TA masked token n-grams, we set $n = 3$ as a lower limit, as one of our intentions is to capture (masked) topic words surrounded by topic-agnostic words, so that $n = 3$ is a minimum limit. As an upper limit, we set $n = 4$ for the same reason mentioned for TA token n-grams.

3.3 Scope of Feature Extraction

In existing AV studies it is often not mentioned which **scope** is considered to extract n-gram-based features. Here, the scope might be the entire text, paragraphs, sentences, clauses, phrases or tokens. Depending on the considered scope, the dimension of the generated feature space may vary which, in turn, may affect the verification results. For example, extracting token n-grams from single sentences would result in a smaller number of features, in contrast to the extraction from the whole text. This is because token n-grams cross sentence boundaries, so that respective cross-sentence features are not taken into account. Despite the smaller number of available features, we have decided, with regard to our AV approach, to extract all n-gram-based features exclusively from the **sentence-level** of the documents. The reason for this is that in practice short text fragments (e. g., social media posts or email text bodies) are often concatenated to obtain a sufficient document length, so that one sentence might not always have a connection to a subsequent sentence. Hence, if we extract n-gram-based features from the entire text, we would erroneously create artificial cross-sentence features that may not occur in texts of a particular author. Note that for feature extraction, we only consider lower case in order to capture all possible case variants (for example, `"The"`, `"the"` or `"THE"`), which can occur especially in informal texts.

4 Verification Method

In this section, we present our AV approach TAVeer[4], which is inspired by the methodology of biometric recognition systems. These aim to recognize individuals, based on a variety of physiological characteristics and behavioral features obtained from the hand, vein, fingerprint, face, eye, ear or voice. Here, the "**E**qual **E**rror **R**ate" (EER) represents a statistic used to show biometric performance in the context of a verification task. Essentially, EER corresponds to a point on a ROC curve where the false acceptance rate is equal to the false rejection rate. Given a questioned document $\mathcal{D}_{\mathcal{U}}$ and a document $\mathcal{D}_{\mathcal{A}}$ from a known author $\mathcal{A}$, the goal of our method is to determine whether $\mathcal{D}_{\mathcal{U}}$ was also written by $\mathcal{A}$. To achieve this goal, TAVeer employs an ensemble of m distance-based classifiers, where each one aims to accept or reject the questioned authorship of $\mathcal{D}_{\mathcal{U}}$. Each classifier is provided with a category of stylistic features extracted from an individual linguistic layer (in each document). In this context, EER serves as a thresholding mechanism, where erroneous verification predictions in either direction are treated equally. This is different from other AV methods as, for example, the approach of Bevendorff et al. [2] that heavily prioritize precision over recall.

TAVeer can essentially be divided into the two phases **training** and **inference**. In the training stage, a model $\mathcal{M}$ has to be "learned" on the basis of a given training corpus $\mathcal{C} = (c_1, c_2, \ldots, c_n)$. Here, each c denotes a verification case, for which the ground truth (`Y` or `N`) is known. In the inference stage, the generated model $\mathcal{M}$ is applied to an unseen verification case in order to accept or reject the questioned authorship. In the following we first describe the preliminaries for TAVeer and then the two phases.

4.1 Preliminaries

Before describing our approach in detail, we first explain what exactly is considered as an input, how this input is represented and on which basic functionality it depends in order to measure the (de)similarity between the documents.

Document Input TAVeer follows the profile-based paradigm that, to our best knowledge, was first described by Potha and Stamatatos [24] in the context of AV. In case that for a known author $\mathcal{A}$ a set of reference documents $\mathbb{D}_{\mathcal{A}} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots\}$ is provided, the idea behind the profile-based approach is to concatenate all documents in $\mathbb{D}_{\mathcal{A}}$ into a single document $\mathcal{D}_{\mathcal{A}}$. Thus, a verification case c is transformed from $(\mathcal{D}_{\mathcal{U}}, \mathbb{D}_{\mathcal{A}})$ to $(\mathcal{D}_{\mathcal{U}}, \mathcal{D}_{\mathcal{A}})$, which represents the document input for TAVeer.

Document Representation As a document representation technique, we consider a *bag-of-features* model, in which all involved features are treated independently from each other. Let $\mathbb{F} = \{F_1, F_2, \ldots F_m\}$ be the m proposed feature categories (cf. Table 1). We define a function $f : \mathbb{D} \times \mathbb{D} \times \mathbb{F} \rightarrow \bigcup_{k \in \mathbb{N}} \mathbb{R}^k \times \mathbb{R}^k$, which transforms $\mathcal{D}_{\mathcal{U}}$ and $\mathcal{D}_{\mathcal{A}}$ according to a given feature category F to two real valued vectors, where k denotes the dimension of the feature space spanned by F. Consider for example F_1

[4] TAVeer stands for "**T**opic-agnostic **A**uthorship **V**erifier based on **e**qual **e**rror **r**ate".

as a feature category, which describes a set of punctuation marks {"-", ";", "?", ...}. Applying f to $\mathcal{D}_{\mathcal{U}}$ and $\mathcal{D}_{\mathcal{A}}$ yields all punctuation marks, that exist in at least one of the documents and adds them to a list $\mathcal{V} = (v_1, v_2, \ldots, v_k)$. Then, two vectors $X = (x_1, x_2, \ldots, x_k)$ and $Y = (y_1, y_2, \ldots, y_k)$ are created, where each x_j and y_j represents the absolute frequency of the corresponding punctuation mark $v_j \in \mathcal{V}$ in each document, respectively. As a final step, we normalize each vector by its *Manhattan* norm $\|\cdot\|_1$, so that all contained features are scaled into the (real) interval $[0, 1]$ and sum up to one. This procedure holds for all m feature categories.

Distance Function To measure the (dis)similarity between two generated feature vectors X and Y, we use a distance function $\text{dist}(X, Y)$. For this, we have chosen the well-known *Manhattan* metric, defined by:

$$\text{dist}(X, Y) = \|X - Y\|_1 = \sum_{r=1}^{k} |X_r - Y_r| \tag{1}$$

which has been used in a number of previous stylometry studies (for example, [1,5]). The *Manhattan metric* benefits from its simplicity and also from the fact that it allows easy interpretation[5] of which specific features have contributed to the prediction.

4.2 Model Learning

Given the training corpus $\mathcal{C}$ and the set of the m feature categories $\mathbb{F} = \{F_1, F_2, \ldots F_m\}$, the objective of this step is to construct a model $\mathcal{M}$, which represents the optimal combination of feature categories obtained on $\mathcal{C}$. In the following, we describe the necessary sub-steps to create $\mathcal{M}$.

Computing Thresholds In this sub-step, we have to compute the individual thresholds $\Theta = (\theta_{F_1}, \theta_{F_2}, \ldots, \theta_{F_m})$ for the m feature categories. Using Equation 1, we calculate for each verification case $c_j = (\mathcal{D}_{\mathcal{A},j}, \mathcal{D}_{\mathcal{U},j}) \in \mathcal{C}$ and each feature category F_i the respective distance $d_{i,j} = \text{dist}(f(\mathcal{D}_{\mathcal{A},j}, \mathcal{D}_{\mathcal{U},j}, F_i))$. As a thresholding technique, we select the *equal error rate* (EER), which describes the point, where the false positives rate is equal to the false negatives rate. Since all corpora used in our experimental setting are balanced, a threshold, which will result in an EER, can be obtained by calculating the median of the distances over all cases in the corpus. Consequently, for all m feature categories, we obtain the corresponding thresholds as follows:

$$\Theta = (\theta_{F_1}, \theta_{F_2}, \ldots, \theta_{F_m}), \text{ with } \theta_{F_i} = \text{median}(d_{i,1}, d_{i,2}, \ldots, d_{i,n}) \tag{2}$$

Note that in case where an exact EER is not feasible (for example, when multiple distance values are equal) the median provides the closest approximation of the EER.

[5] We refer the interested reader to our extended version of this paper [11], in which we explain in detail how the interpretation can be performed.

Similarity Function The introduced distance function (cf. Equation 1) allows us to compute distances between a pair of two feature vectors X and Y. However, the resulting distances are not calibrated with respect to the individual thresholds from the previous sub-step. Therefore, we designed a similarity function $\text{sim}(\cdot)$ that considers as an input a distance d, a threshold θ_F and the upper bound d_{max} of the provided distance function (in our case, the *Manhattan metric*). Recall that in the context of our approach, all feature vectors are normalized using the *Manhattan* norm $\|\cdot\|_1$. Consequently, all features in each vector sum up to 1. Based on this fact, the lower and upper bound of $\text{dist}(X, Y)$ can be calculated by

$$\mathbf{0} \leq \|X - Y\|_1 \leq \|X\|_1 + \|Y\|_1 = \mathbf{2}$$

such that $d_{max} = 2$ holds. An important requirement regarding our similarity function is that the resulting score s is calibrated in a way that 0.5 represents the decision boundary. One possible definition for a function $\text{sim}(\cdot)$ that transforms a distance d into the range $[0, 1]$ and simultaneously calibrates the resulting similarity score s with respect to this "natural" decision boundary is:

$$\text{sim}(d, d_{max}, \theta_F) = \begin{cases} 1 - \frac{d}{2\theta_F}, & \text{if } d \leq \theta_F, \\ \frac{1}{2} - \frac{d - \theta_F}{2(d_{max} - \theta_F)}, & \text{otherwise} \end{cases} \tag{3}$$

Figure 1 illustrates the behavior of $\text{sim}(\cdot)$ with respect to the lower and upper bound of the *Manhattan metric*. Note that by considering d_{max} as a variable parameter, we can

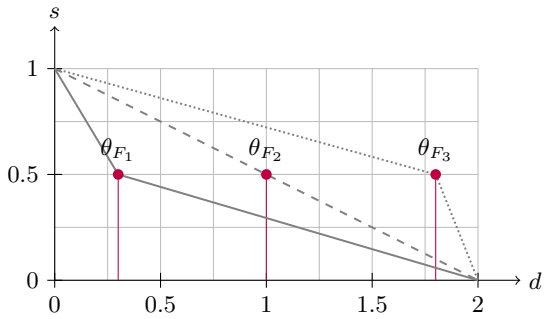

Figure 1. Behavior of the proposed similarity function with respect to the given distance $\boldsymbol{d}$ for $\boldsymbol{d_{max} = 2}$ and three sample thresholds $\boldsymbol{\theta_{F_1} = 0.3, \theta_{F_2} = 1}$ and $\boldsymbol{\theta_{F_3} = 1.8}$.

easily substitute the *Manhattan metric* with any other distance function, as long as its respective upper bound d_{max} is known. Furthermore, it should be noticed that any other definition for $\text{sim}(\cdot)$ that also fulfills the same requirement can be used instead.

Classification Function The similarity function $\text{sim}(\cdot)$ from the previous sub-step can already calculate a calibrated similarity value for a given distance d and a threshold for

a **single** feature category. However, the idea behind TAVeer is to determine whether a questioned authorship between two documents holds based on **multiple** feature categories. Let $\mathbb{F}_\Theta = \{(F_i, \theta_{F_i}) | i \in \{1, 2, \ldots, m\}\}$ denote a set, which comprises pairs of feature categories and their associated thresholds and $\mathcal{P}(\mathbb{F}_\Theta)$ the power set (without the empty set) holding all possible combinations of these pairs. We denote a single $\mathcal{E} \in \mathcal{P}(\mathbb{F}_\Theta)$ by the term **ensemble**. Furthermore, we denote an ensemble comprising a single pair $\{(F, \theta_F)\} \subseteq \mathcal{E}$ as an **atomic ensemble**. To compute a similarity value with respect to $\mathcal{E}$, we define an aggregated similarity function $\text{sim}_\mathcal{E}(\cdot)$ as follows:

$$\begin{aligned} &\text{sim}_\mathcal{E}(\mathcal{D}_\mathcal{U}, \mathcal{D}_\mathcal{A}, d_{max}, \mathcal{E}) = \text{median}\,(\mathcal{S})\,, \text{with} \\ &\mathcal{S} = \{\text{sim}(\text{dist}(f(\mathcal{D}_\mathcal{U}, \mathcal{D}_\mathcal{A}, F)), d_{max}, \theta_F) | (F, \theta_F) \in \mathcal{E}\} \end{aligned} \tag{4}$$

To obtain a binary prediction (Y/N) for a single verification case c based on $\text{sim}_\mathcal{E}(\cdot)$, we further define a classification function:

$$\text{clf}(\mathcal{D}_\mathcal{U}, \mathcal{D}_\mathcal{A}, d_{max}, \mathcal{E}) = \begin{cases} \texttt{Y}, & \text{if } \text{sim}_\mathcal{E}(\mathcal{D}_\mathcal{U}, \mathcal{D}_\mathcal{A}, d_{max}, \mathcal{E}) > 0.5 \\ \texttt{N}, & \text{otherwise} \end{cases} \tag{5}$$

Selecting Optimal Ensemble In this last sub-step, the goal is to determine the optimal ensemble on the basis of the training corpus $\mathcal{C}$, which will serve as the model $\mathcal{M}$ for the inference stage. To achieve this goal, we use Equation 5 to classify all verification cases $c_1, c_2, \ldots, c_n$ in $\mathcal{C}$ for each possible ensemble $\mathcal{E} \in \mathcal{P}(\mathbb{F}_\Theta)$. As a result, we obtain $|\mathcal{P}(\mathbb{F}_\Theta)|$ predictions for each c_i. Based on the predictions and the ground truth provided for $\mathcal{C}$, we can now calculate the accuracies for each ensemble to find the optimal one that will represent $\mathcal{M}$. One way to obtain an optimal ensemble would be to select the one that leads to a maximum accuracy on $\mathcal{C}$. In practice, however, this approach is not always reasonable as several ensembles can share the maximum accuracy. For this reason, we decided to consider additional criteria to obtain an optimal ensemble. Based on the power set $\mathcal{P}(\mathbb{F}_\Theta)$, we sort all the resulting ensembles one by one according to the following three criteria (each in descending order):

1. Accuracy of an ensemble $\mathcal{E}$ (calculated for $\mathcal{C}$)
2. Number of feature categories an ensemble $\mathcal{E}$ contains
3. Median accuracy regarding all atomic ensembles in $\mathcal{E}$ (calculated for $\mathcal{C}$)

From here, it is unlikely that multiple ensembles share the same ranking regarding these criteria. Finally, we select the first ensemble from the sorted list, which will serve as the final model $\mathcal{M}$.

4.3 Inference

In contrast to the training phase, the inference phase is much more compact. Here, TAVeer consumes the resulting model $\mathcal{M}$ from the training phase and performs the following steps to classify an unseen verification case $c_? = (\mathcal{D}_\mathcal{U}, \mathcal{D}_\mathcal{A})$. Using Equation 4, TAVeer first computes the similarity value $s_?$ between the unknown and known documents $\mathcal{D}_\mathcal{U}$ and $\mathcal{D}_\mathcal{A}$. Afterwards, a binary prediction regarding the questioned authorship

of $\mathcal{D}_{\mathcal{U}}$ is obtained by comparing $s_?$ against the decision boundary 0.5 (cf. Equation 5). In case that $s_? > 0.5$ holds, $c_?$ is classified as `Y` ($\mathcal{D}_{\mathcal{U}}$ and $\mathcal{D}_{\mathcal{A}}$ are assumed to be written by the same author), otherwise as `N` (both documents are probably written by different authors).

5 Experimental Evaluation

This section gives a brief description of our experimental evaluation. At the time we developed TAVeer, we had no access to the official test corpus and the respective ground truth of the underlying verification cases. To train and evaluate our approach, we therefore have split the official training[6] data set provided by the PAN organizers into a training and validation corpus. In the following, we first explain how the initial training data set was partitioned, summarize its key statistics and mention several relevant observations we have made in regard to the verification cases in the corpus. Afterwards, we describe which alternative performance measure we have chosen to evaluate TAVeer on the validation corpus. Finally, we present the results on this corpus as well as on the official test corpus.

5.1 Corpora

From the given official training data set, we reserved a fraction of 5,000 cases to train[7] and 47,590 cases to evaluate TAVeer. Table 3 summarizes the statistics of both partitions. During our examination of the documents within the corpus, we made some observations worth mentioning. In a number of verification cases, the known and unknown are written in different languages. For example, within the verification cases:

```
2225c14b-e691-5c6b-833f-0eea70a8be9c
a5bf996f-0fd1-57c0-9953-5c99155e4a47
831efc2b-edab-56a6-8a38-a8b18273363f
```

one document is written in English while the other is written in Spanish, Swedish and French, respectively. Within the case:

```
33c96c88-acd5-503f-ac71-7397a277d144
```

both the unknown and known document are identical and in the case:

```
2a7758a1-1f08-503d-82b1-dfdf8e928560
```

one document contains a valid natural language text, while the other one contains almost entirely repetitions of the same word. Besides these manual inspected verification cases, we further performed an automated analysis with regard to all documents contained in the training and validation corpora. Here, we noticed that a large fraction of the documents contain an excessive number of quotes. While trying to remove these

[6] Note that we used the "small" version of the official training corpus.

[7] Note that the submitted version of our approach was only trained on this partition. In other words, we have **not** retrained TAVeer on the entire training data set.

quotes, we found that they made up about half of the texts and also, that apostrophes and quotation marks have been normalized by the same character `"`, which further complicated to remove the quotes. In view of these observations, we have left the documents in their original form, so that no cleaning has been carried out at all (this also applies to the test corpus).

Corpus $\mathcal{C}$	$\lvert\mathcal{C}\rvert$	**Distribution (`Y`/`N`)**	$\lvert\mathbb{D}_{\mathcal{A}}\rvert$	**avg**$\lvert\mathcal{D}_{\mathcal{A}}\rvert$	**avg**$\lvert\mathcal{D}_{\mathcal{U}}\rvert$
$\mathcal{C}_{\text{PAN}}$ (train)	5,000	2,500 / 2,500	1	21,396	21,392
$\mathcal{C}_{\text{PAN}}$ (validation)	47,590	25,323 / 22,267	1	21,452	21,439

Table 3. Key statistics for the training and validation partitions. Notation: $|\mathcal{C}|$ denotes the number of verification cases in each corpus $\mathcal{C}$, while $|\mathbb{D}_{\mathcal{A}}|$ denotes the number of the known documents. The average character length of $\mathcal{D}_{\mathcal{U}}$ and $\mathcal{D}_{\mathcal{A}}$ (concatenation of all documents in $\mathbb{D}_{\mathcal{A}}$) is denoted by avg$|\mathcal{D}_{\mathcal{U}}|$ and avg$|\mathcal{D}_{\mathcal{A}}|$, respectively.

5.2 Performance Measures

To assess the performance of our approach on the validation corpus, we have selected balanced accuracy (BAC) as an alternative performance measure. Despite its robustness and suitability especially for imbalanced corpora, BAC has not yet been considered in the field of AV, to the best of our knowledge. In contrast to F_1 and the newly proposed measure $F_{0.5u}$ [2], BAC considers all four confusion matrix outcomes: true positives (TP), false negatives (FN), false positives (FP) and true negatives (TN). This is preferable[8] in realistic forensic cases where two opposing goals are faced:

1. verify that an alleged authorship is indeed correct, or
2. falsify an alleged authorship correctly

so that both TP and TN can be measured reliably at the same time. BAC is defined by the arithmetic mean of **sensitivity** = true positive rate (TPR) and **specificity** = false positive rate (FPR):

$$\text{BAC} = \frac{1}{2}\Big(\text{TPR} + \text{FPR}\Big) = \frac{1}{2}\Big(\frac{\text{TP}}{\text{TP + FN}} + \frac{\text{TN}}{\text{TN + FP}}\Big)$$

When dealing with imbalanced corpora, we have observed that BAC is easier to interpret than other recommended measures such as Cohen's κ [10]. For balanced corpora, on the other hand, BAC offers another benefit making it a reliable performance measure as it is equal to ordinary accuracy. If we consider an AV method that (due to a weak calibration) predicts nothing but `Y` (same-author) or `N` (different-author), the resulting BAC value will always be 0.5. When using F_1, which behaves **asymmetric** regarding one-sided `Y`- and `N`-predictions, the resulting score is either $\frac{2}{3} \approx 0.66$ or 0, respectively.

[8] This is at least true for the real-world forensic cases we have worked on in our research department at Fraunhofer SIT.

In addition to BAC, we also report the confusion matrix outcomes to allow a better comparability regarding the results made by TAVeer, as well as the four measures (AUC, c@1, $F_{0.5u}$ and F_1) considered by the PAN organizers for the official evaluation results.

5.3 Results

After training TAVeer on the partition with the 5,000 verification cases, we applied the trained model to the imbalanced validation corpus comprising 47,590 cases. The results for the validation corpus are shown in Table 4, while the official results[9] with regard to the test corpus are listed in Table 5. A comparison of the results of the validation and the test corpora shows that TAVeer can generalize well, where only minimal losses can be observed for the test corpus. However, since the PAN organizers did not report the four confusion matrix outcomes for the test corpus, we cannot infer from the single number metrics more fine-grained information regarding the individual classification predictions of TAVeer or the other submitted AV approaches. Furthermore, we cannot provide any analysis regarding the test corpus, since at the time this paper was written we have no access to it.

BAC	AUC	c@1	$F_{0.5u}$	F_1	TP	FN	FP	TN
0.819	0.897	0.818	0.838	0.824	20,270	5,053	3,624	18,643

Table 4. Evaluation results of TAVeer and the ten selected baseline methods. Bold and underlined values represent the best and second best results.

Since we cannot make any statement regarding the test corpus, we have decided to use two self-compiled corpora $\mathcal{C}_{\text{Reddit}}$ and $\mathcal{C}_{\text{Amazon}}$ in order to get a better understanding regarding the cross-domain capability of TAVeer. $\mathcal{C}_{\text{Reddit}}$ contains (partially very colloquial) comments from the well-known Reddit platform, while $\mathcal{C}_{\text{Amazon}}$ contains product reviews from the Amazon platform. Both corpora, which differ in topic and genre, are described in detail in our paper [12] along with their respective corpus statistics. In what follows, we therefore only focus on the cross-domain experiment. The question we are seeking to answer is to what extent a model $\mathcal{M}_X$ learned on a training corpus from a domain $\mathcal{X}$ can be applied to a test corpus from a domain $\mathcal{Y}$ (with $\mathcal{X} \neq \mathcal{Y}$) and vice versa.

Using the procedure described in Section 4.2, we first learn the two models $\mathcal{M}_{\text{Reddit}}$ and $\mathcal{M}_{\text{Amazon}}$ (cf. Table 7) on the training partitions of the corpora $\mathcal{C}_{\text{Reddit}}$ and $\mathcal{C}_{\text{Amazon}}$. Based on $\mathcal{M}_{\text{Reddit}}$, we then apply TAVeer to the test partition of $\mathcal{C}_{\text{Amazon}}$. Afterwards, we apply $\mathcal{M}_{\text{Amazon}}$ to the test partition of $\mathcal{C}_{\text{Reddit}}$. The results are shown in Table 6. If we focus on the performance deviations between the models applied to the original and cross-domain corpora, we can see in this table a slight loss of −0.005 in terms of BAC

[9] The results have been taken from the PAN website https://pan.webis.de/clef20/pan20-web/author-identification.html

Rank	Team	Training dataset	AUC	c@1	$F_{0.5u}$	F_1	Overall
1	Boenninghoff20	large	**0.969**	**0.928**	**0.907**	**0.936**	**0.935**
2	Weerasinghe20	large	<u>0.953</u>	0.880	<u>0.882</u>	0.891	<u>0.902</u>
3	Boenninghoff20	small	0.940	<u>0.889</u>	0.853	<u>0.906</u>	0.897
4	Weerasinghe20	small	0.939	0.833	0.817	0.860	0.862
5	Halvani20b	small	0.878	0.796	0.819	0.807	0.825
6	Kipnis20	small	0.866	0.801	0.815	0.809	0.823
7	Araujo20	small	0.874	0.770	0.762	0.811	0.804
8	Niven20	small	0.795	0.786	0.842	0.778	0.800
9	Gagala20	small	0.786	0.786	0.809	0.800	0.796
10	Araujo20	large	0.859	0.751	0.745	0.800	0.789
11	Baseline (naive)	small	0.780	0.723	0.716	0.767	0.747
12	Baseline (compression)	small	0.778	0.719	0.703	0.770	0.742
13	Ordonez20	large	0.696	0.640	0.655	0.748	0.685
14	Faber20	small	0.293	0.331	0.314	0.262	0.300

Table 5. Official evaluation results for the PAN 2020 test corpus with regard to the "large" and "small" training data sets. Bold and underlined values represent the best and second best results.

and a small gain of +0.002 in terms of AUC for the $\mathcal{C}_{\text{Reddit}}$ test partition. Similarly, for the test partition of $\mathcal{C}_{\text{Amazon}}$, we can observe a slightly greater loss of −0.032 and −0.012 in terms of BAC and AUC, respectively.

The reason for the small deviations can be explained by the fact that the majority of the feature categories (more precisely, F_1, F_2, F_4, F_6 and F_{11}) are present in both models $\mathcal{M}_{\text{Reddit}}$ and $\mathcal{M}_{\text{Amazon}}$, as can be seen in Table 7. Furthermore, their respective thresholds are very similar to each other. Consequently, both models are interchangeable without major performance losses, so that (at least on these corpora) TAVeer can be considered robust with respect to the different domains.

Corpus	Model	BAC	AUC	c@1	$F_{0.5u}$	F_1	TP	FN	FP	TN
$\mathcal{C}_{\text{Reddit}}$ (test)	$\mathcal{M}_{\text{Reddit}}$	0.806	0.861	0.806	0.821	0.796	455	145	88	512
$\mathcal{C}_{\text{Reddit}}$ (test)	$\mathcal{M}_{\text{Amazon}}$	0.801	0.863	0.801	0.810	0.794	462	138	101	499
$\mathcal{C}_{\text{Amazon}}$ (test)	$\mathcal{M}_{\text{Amazon}}$	0.842	0.912	0.842	0.851	0.838	982	218	161	1039
$\mathcal{C}_{\text{Amazon}}$ (test)	$\mathcal{M}_{\text{Reddit}}$	0.810	0.900	0.810	0.813	0.808	959	241	216	984

Table 6. Cross-domain evaluation results for our two self-compiled corpora $\mathcal{C}_{\text{Reddit}}$ and $\mathcal{C}_{\text{Amazon}}$. Note that since both corpora are balanced, the BAC and c@1 values are equal.

6 Conclusion and Future Work

We have presented a simple but effective distance-based authorship verification (AV) approach called TAVeer to the AV 2020 shared task of the PAN competition, where the task was to determine for a pair of documents if both texts were written by the

Corpus	(F_1, θ_{F_1})	(F_2, θ_{F_2})	(F_4, θ_{F_4})	(F_6, θ_{F_6})	(F_7, θ_{F_7})	(F_8, θ_{F_8})	(F_9, θ_{F_9})	$(F_{10}, \theta_{F_{10}})$	$(F_{11}, \theta_{F_{11}})$
$\mathcal{C}_{\text{Reddit}}$	$(F_1, 0.343)$	$(F_2, 0.757)$	$(F_4, 1.181)$	$(F_6, 0.641)$		$(F_8, 1.956)$	$(F_9, 1.996)$	$(F_{10}, 1.671)$	$(F_{11}, 1.869)$
$\mathcal{C}_{\text{Amazon}}$	$(F_1, 0.349)$	$(F_2, 0.801)$	$(F_4, 1.108)$	$(F_6, 0.680)$	$(F_7, 1.622)$				$(F_{11}, 1.862)$

Table 7. Model analysis: Each row (starting with column two) represents a model $\mathcal{M}$ learned on the respective training partition. Recall that F_i represents a feature category and θ_{F_i} its corresponding threshold.

same author. Our approach, which we call TAVeer, relies solely on topic-agnostic feature categories based on punctuation marks, function words, contractions, transitional phrases as well as several subclasses of verbs and adverbs. By this, the method differs from many existing approaches that rely on implicitly defined feature categories such as character n-grams. Using such feature categories, in particular, in the context of AV is problematic, as one has no control over the features that are indeed captured. In the worst case, the prediction of an AV method may be based on topic-related words rather than on stylistic features, so that the method will miss its true purpose. The core of TAVeer is a distance function (Manhattan metric), which in combination with a thresholding procedure (based on equal error rate) acts as the underlying classifier.

To assess our approach, we have split the training data set into a training and validation set, where for the former only 5,000 verification cases were used (in other words, less than 10% of the entire data set). This model was submitted for the final evaluation on the official test set. From the official evaluation results and those obtained on our validation corpus, it can be concluded that TAVeer is able to generalize well across both corpora, with minimal losses on the test corpus. Besides the official train and test corpora, we have further performed a cross-domain experiment regarding two self-compiled corpora. In this context, we have demonstrated that TAVeer performs robustly even though the trained models and the test corpora come from two different domains.

Nevertheless, our AV method leaves room for further improvements. Currently, TAVeer does not take into account misspelled words, which can lead to a loss of potentially relevant features, especially in connection with informal texts. We therefore leave for future work the investigation of effective possibilities to semantically match misspelled words with respect to their common entity. One idea, for example, is to use back-translation services that can handle difficult spelling mistakes, which cannot be corrected by standard spell checkers. Another direction for future work is to investigate alternative feature categories not yet been considered in this paper. In this context, one idea is to experiment with interjections (e. g., `"lol"` or `"aha"`) or topic-agnostic abbreviations (for example, `"e.g."` or `"etc."`), which represent important idiosyncratic stylistic markers.

7 Acknowledgments

This research work has been funded by the German Federal Ministry of Education and Research and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

References

1. Ahmed, H.: The Role of Linguistic Feature Categories in Authorship Verification. Procedia Computer Science **142**, 214 – 221 (2018), arabic Computational Linguistics
2. Bevendorff, J., Stein, B., Hagen, M., Potthast, M.: Generalizing Unmasking for Short Texts. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 654–659. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
3. Binongo, J.N.G.: Who Wrote the 15th Book of Oz? An Application of Multivariate Analysis to Authorship Attribution. CHANCE **16**(2), 9–17 (2003)
4. Brocardo, M.L., Traore, I., Saad, S., Woungang, I.: Authorship Verification for Short Messages Using Stylometry. In: 2013 International Conference on Computer, Information and Telecommunication Systems (CITS). pp. 1–6 (May 2013)
5. Burrows, J.: Delta: a Measure of Stylistic Difference and a Guide to Likely Authorship. Literary and Linguistic Computing **17**(3), 267–287 (2002)
6. Castro Castro, D., Adame Arcia, Y., Pelaez Brioso, M., Muñoz Guillena, R.: Authorship Verification, Average Similarity Analysis. In: Proceedings of the International Conference Recent Advances in Natural Language Processing. pp. 84–90. INCOMA Ltd. Shoumen, BULGARIA (2015)
7. Chandrasekaran, R., Manimannan, G.: Use of Generalized Regression Neural Network in Authorship Attribution. International Journal of Computer Applications **62**(4), 7–10 (January 2013)
8. De Smedt, T., Daelemans, W.: Pattern for Python. J. Mach. Learn. Res. **13**(1), 2063–2067 (Jun 2012)
9. Gamon, M.: Linguistic Correlates of Style: Authorship Classification with Deep Linguistic Analysis Features. In: Proceedings of Coling 2004. pp. 611–617. International Conference on Computational Linguistics (August 2004)
10. Halvani, O., Graner, L.: Rethinking the Evaluation Methodology of Authorship Verification Methods. In: Bellot, P., Trabelsi, C., Mothe, J., Murtagh, F., Nie, J.Y., Soulier, L., SanJuan, E., Cappellato, L., Ferro, N. (eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction. pp. 40–51. Springer International Publishing (2018)
11. Halvani, O., Graner, L., Regev, R.: A Step Towards Interpretable Authorship Verification. CoRR **abs/2006.12418** (2020)
12. Halvani, O., Graner, L., Regev, R.: TAVeer: An Interpretable Topic-Agnostic Authorship Verification Method. In: Volkamer, M., Wressnegger, C. (eds.) ARES 2020: The 15th International Conference on Availability, Reliability and Security, Virtual Event, Ireland, August 25-28, 2020. pp. 41:1–41:10. ACM (2020)
13. Halvani, O., Winter, C., Graner, L.: Assessing the Applicability of Authorship Verification Methods. In: Proceedings of the 14th International Conference on Availability, Reliability and Security, ARES 2019, Canterbury, UK, August 26-29, 2019. pp. 38:1–38:10. ACM (2019)
14. Jankowska, M., Milios, E.E., Keselj, V.: Author Verification Using Common N-Gram Profiles of Text Documents. In: Hajic, J., Tsujii, J. (eds.) COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland. pp. 387–397. ACL (2014)
15. Juola, P., Noecker, J., Stolerman, A., Ryan, M., Brennan, P., Greenstadt, R.: Towards Active Linguistic Authentication. In: Peterson, G., Shenoi, S. (eds.) Advances in Digital Forensics IX. pp. 385–398. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
16. Juola, P., Stamatatos, E.: Overview of the Author Identification Task at PAN 2013. In: Working Notes for CLEF 2013 Conference, Valencia, Spain, September 23-26, 2013 (2013)

17. Khonji, M., Iraqi, Y.: A Slightly-Modified GI-Based Author-Verifier with Lots of Features (ASGALF). In: Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014. pp. 977–983 (2014)
18. Kocher, M., Savoy, J.: Unine at CLEF 2015 author identification: Notebook for PAN at CLEF 2015. In: CLEF (Working Notes). CEUR Workshop Proceedings, vol. 1391. CEUR-WS.org (2015)
19. Kocher, M., Savoy, J.: A Simple and Efficient Algorithm for Authorship Verification. Journal of the Association for Information Science and Technology **68**(1), 259–269 (2017)
20. Koppel, M., Schler, J., Argamon, S.: Computational Methods in Authorship Attribution. JASIST **60**(1), 9–26 (2009)
21. Koppel, M., Winter, Y.: Determining if Two Documents are Written by the Same Author. JASIST **65**(1), 178–187 (2014)
22. Neal, T.J., Sundararajan, K., Woodard, D.L.: Exploiting Linguistic Style as a Cognitive Biometric for Continuous Verification. In: 2018 International Conference on Biometrics, ICB 2018, Gold Coast, Australia, February 20-23, 2018. pp. 270–276. IEEE (2018)
23. Pavelec, D., Oliveira, L.S., Justino, E.J.R., Batista, L.V.: Using conjunctions and adverbs for author verification. J. UCS **14**(18), 2967–2981 (2008)
24. Potha, N., Stamatatos, E.: A Profile-Based Method for Authorship Verification. In: Artificial Intelligence: Methods and Applications: 8th Hellenic Conference on AI, SETN 2014, Ioannina, Greece, May 15–17, 2014. Proceedings. pp. 313–326. Springer International Publishing (2014)
25. Potha, N., Stamatatos, E.: An Improved Impostors Method for Authorship Verification. In: Jones, G.J.F., Lawless, S., Gonzalo, J., Kelly, L., Goeuriot, L., Mandl, T., Cappellato, L., Ferro, N. (eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction - 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11-14, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10456, pp. 138–144. Springer (2017)
26. Potha, N., Stamatatos, E.: Improved Algorithms for Extrinsic Author Verification. Knowledge and Information Systems (Oct 2019)
27. Rao, O.S., Raju, N.V.G., Kumar, V.V.: Authorship attribution on imbalanced english editorial corpora. International Journal of Computer Applications **169**(1), 44–47 (Jul 2017)
28. Seidman, S.: Authorship Verification Using the Impostors Method Notebook for PAN at CLEF 2013. In: Working Notes for CLEF 2013 Conference , Valencia, Spain, September 23-26, 2013. (2013)
29. Stamatatos, E., Daelemans, W., Verhoeven, B., Stein, B., Potthast, M., Juola, P., Sánchez-Pérez, M.A., Barrón-Cedeño, A.: Overview of the Author Identification Task at PAN 2014. In: Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15–18, 2014. pp. 877–897 (2014)
30. Stolerman, A.: Authorship Verification. Ph.D. thesis (2015), uMI Dissertations Publishing 2015
31. Varela, P., Justino, E., Soares de Oliveira, L.: Verbs and Pronouns for Authorship Attribution (01 2010)
32. Veenman, C.J., Li, Z.: Authorship Verification with Compression Features. In: Working Notes for CLEF 2013 Conference , Valencia, Spain, September 23–26, 2013 (2013)
33. Zhao, Y., Zobel, J.: Effective and Scalable Authorship Attribution Using Function Words. In: Lee, G., Yamada, A., Meng, H., Myaeng, S. (eds.) Information Retrieval Technology. Lecture Notes in Computer Science, vol. 3689, pp. 174–189. Springer Berlin Heidelberg (2005)
34. Zhao, Y., Zobel, J., Vines, P.: Using relative entropy for authorship attribution. In: Ng, H.T., Leong, M.K., Kan, M.Y., Ji, D. (eds.) Information Retrieval Technology. pp. 92–105. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)